

Closed-Form Training of Conditional Random Fields for Large Scale Image Segmentation

Alexander Kolesnikov

IST Austria

akolesnikov@ist.ac.at

Vittorio Ferrari

University of Edinburgh

vittorio.ferrari@ed.ac.uk

Matthieu Guillaumin

ETH Zürich

guillaumin@vision.ee.ethz.ch

Christoph H. Lampert

IST Austria

chl@ist.ac.at

Abstract

We present *LS-CRF*, a new method for very efficient large-scale training of Conditional Random Fields (CRFs). It is inspired by existing closed-form expressions for the maximum likelihood parameters of a generative graphical model with tree topology. *LS-CRF* training requires only solving a set of independent regression problems, for which closed-form expression as well as efficient iterative solvers are available. This makes it orders of magnitude faster than conventional maximum likelihood learning for CRFs that require repeated runs of probabilistic inference. At the same time, the models learned by our method still allow for joint inference at test time.

We apply *LS-CRF* to the task of semantic image segmentation, showing that it is highly efficient, even for loopy models where probabilistic inference is problematic. It allows the training of image segmentation models from significantly larger training sets than had been used previously. We demonstrate this on two new datasets that form a second contribution of this paper. They consist of over 180,000 images with figure-ground segmentation annotations. Our large-scale experiments show that the possibilities of CRF-based image segmentation are far from exhausted, indicating, for example, that semi-supervised learning and the use of non-linear predictors are promising directions for achieving higher segmentation accuracy in the future.

1. Introduction

Conditional random fields (CRFs) [19] have emerged as powerful tools for modeling several interacting objects, such as parts of bodies, or pixels in an image. As a consequence, they have found multiple applications in computer vision, in particular in *human pose estimation*, *action recog-*

nition and *semantic image segmentation* [21]. However, the increased modeling capacity of conditional random fields comes at a computational price: exact training of CRFs requires repeated runs of probabilistic inference. For simple chain- or tree-structured models this is possible but computationally expensive. For loopy models, as they are dominant in image segmentation, efficient exact algorithms are provably intractable¹. Therefore, an important problem in machine learning and also computer vision research is to find methods for fast *approximate* training of loopy CRFs.

In this work we propose a new technique for fast approximate training of CRFs with up to pairwise terms. We call it *LS-CRF*, where the *LS* stands both for *least squares* and *large scale*. We derive *LS-CRF* in Section 2 from existing closed-form expressions for the maximum likelihood parameters of a non-conditional probability distribution when the underlying graphical model is tree-shaped. We adapt these to the situation of a conditional probability distributions modelled by a CRF, obtaining a training problem that requires only solving several suitably constructed regression problems. For the most common linear parametrization of the CRF energy, these can be solved efficiently using either classical closed-form expressions, or using iterative numerical techniques, such as the conjugate gradient method, or simple stochastic gradient descent. However, we can also opt for a non-linear parameterization, leading to increased expressive power while still staying computationally feasible. Overall, we obtain a CRF training algorithm that is

- **efficient & scalable** (training sets can have 100,000 images or more, subproblems can be solved independently, each subproblem can be parallelized itself)
- **flexible** (we can, e.g., incorporate non-linearities, per-sample weights, or compensate for class imbalance),

¹See Section 3 for the exact characterization of this statement.

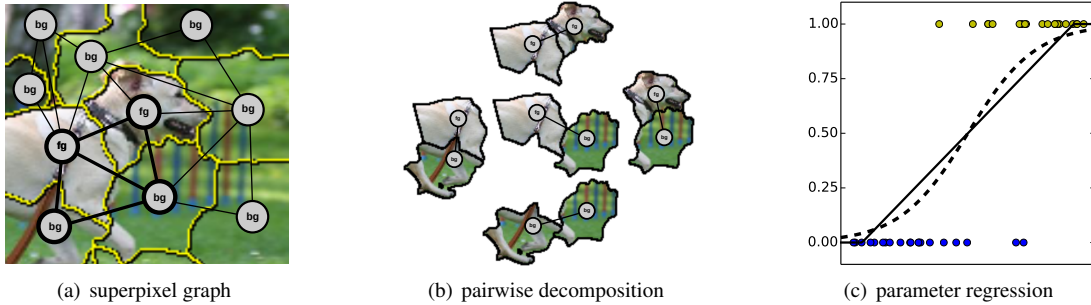


Figure 1: Schematic illustration of LS-CRF for image segmentation – training phase: (a) we are given images with predefined graph structure (here based on superpixels) and per-node annotation (here: **fg/bg**), (b) we form training subproblems from all edges in the graph (shown for bold subgraph), (c) for each label combination, we train a linear (solid line) or nonlinear (dashed line) regressor to predict the label combination’s conditional probability.

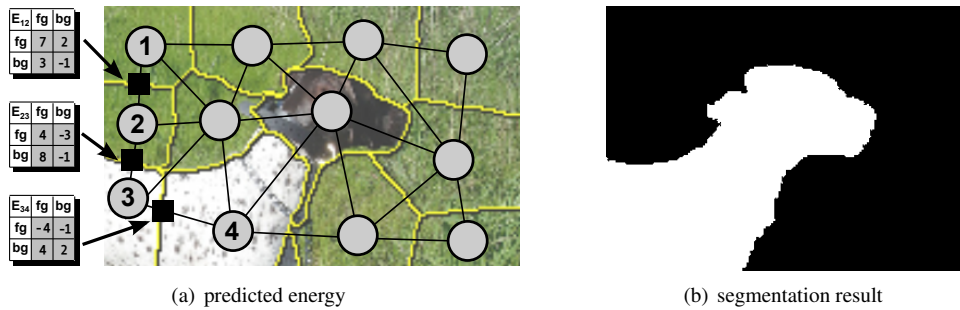


Figure 2: Schematic illustration of LS-CRF for image segmentation – prediction phase: (a) for a new image we use the regressors to predict an energy table for each pairwise term (visualized on three edges between four nodes), (b) The energy function defines a probability distribution which yields a segmentation (here: by MAP prediction).

- **easy to implement** (e.g., the closed-form expressions can be solved in a few lines of Matlab or Python code).

While LS-CRF is a general-purpose training method, it is particularly suitable for the type of CRFs that occur in computer vision applications, such as image segmentation, where: 1) the underlying graph is loopy, 2) all variables are of the same “type” (e.g. pixels or superpixels), 3) each variable takes values in a rather small label set, 4) many training examples are available.

Figures 1 and 2 illustrate LS-CRF for semantic image segmentation. A formal definition and justification are given in Section 2. Our contribution lies in the steps 1(b), 1(c) and 2(a), which we explain in detail in Section 2.

We demonstrate the usefulness of our method by applying it to several image segmentation tasks, both in a semantic multi-class setup as well as a binary figure–ground situation. Our goal in this is to pave the road towards truly large-scale experiments in the area of image segmentation. Since current datasets rarely consists of more than a few hundred images, we created two new datasets of together over 180,000 images. Some of these have manually created annotation, for others, we provide segmentation masks created in a semi-automatic way. By making these publicly

available we hope to stimulate more research in two directions: large-scale CRF training and large-scale (potentially semi-supervised) image segmentation.

We report results of LS-CRF in Section 4 on this dataset as well as on the Stanford backgrounds dataset, comparing our results to baseline techniques in terms of training speed as well as segmentation accuracy.

2. Inference-Free CRF Training

We first reiterate the classical construction of maximum likelihood parameter estimation for non-conditional probability distribution with tree-shaped independence relations (see [30, page 150]). For m variables, y_1, \dots, y_m , where each y_s can take values in label set $\mathcal{L} = \{1, \dots, r\}$, let \mathcal{Y} be the set of all possible joint labelings, i.e. $\mathcal{Y} = \mathcal{L}^m$, and let $P_\theta(y)$ for $y = (y_1, \dots, y_m)$ be a joint probability distribution over such labelings. We assume the structure of an undirected graphical model with underlying graph $G = (V, \mathcal{E})$ with $V = \{1, \dots, m\}$ and edge set $\mathcal{E} \subset V \times V$,

such that $P_\theta(y) \propto \exp(-E_\theta(y))$ for an energy function

$$E_\theta(y) = \sum_{s \in V} \sum_{j \in \mathcal{L}} \theta_{s;j} \mathbb{I}[y_s = j] + \sum_{(s,t) \in \mathcal{E}} \sum_{(j,k) \in \mathcal{L} \times \mathcal{L}} \theta_{st;jk} \mathbb{I}[y_s = j \wedge y_t = k], \quad (1)$$

where $\mathbb{I}[P] = 1$ if the predicate P is true, and 0 otherwise.

The distribution P_θ is fully determined by its parameter vector θ that consists of the unary and pairwise parameters $\theta_{s;j} \in \mathbb{R}$ for all $s \in V$ and $j \in \mathcal{L}$, and $\theta_{st;jk} \in \mathbb{R}$ for all $(s, t) \in \mathcal{E}$ and $(j, k) \in \mathcal{L} \times \mathcal{L}$, respectively.

To *learn* such a distribution means to estimate an unknown θ from a set of i.i.d. samples (y^1, \dots, y^T) using the maximum likelihood principle. It is a classical but rarely used fact that for tree-shaped graphs this step is possible in closed form. Let

$$\hat{\mu}_{s;j} = \frac{1}{T} \sum_{i=1}^T \mathbb{I}[y_s^i = j], \quad \hat{\mu}_{st;jk} = \frac{1}{T} \sum_{i=1}^T \mathbb{I}[y_s^i = j \wedge y_t^i = k], \quad (2)$$

be the empirical estimates of the single and pairwise variable marginal probabilities for all $s \in V$ and $j \in \mathcal{L}$, and for all $(s, t) \in \mathcal{E}$ and $(j, k) \in \mathcal{L} \times \mathcal{L}$. Then

$$\hat{\theta}_{s;j} = \log \hat{\mu}_{s;j}, \quad \hat{\theta}_{st;jk} = \log \frac{\hat{\mu}_{st;jk}}{\hat{\mu}_{s;j} \hat{\mu}_{t;k}} \quad (3)$$

maximum likelihood parameter estimate, as can be checked by a straight-forward computation [30]. This estimate is *consistent*, i.e. $P_{\hat{\theta}}(y)$ converges to $P_\theta(y)$ for $T \rightarrow \infty$.

2.1. Conditional Distributions (LS-CRF Training)

The goal of this work is to generalize the above closed-form expression into a procedure for inference-free conditional random field (CRF) learning. The main problem we have to solve is that above section applies only to ordinary fixed probability distribution, whereas a CRFs encode conditional distributions, i.e. a different distribution $P(y|x)$ for each x from a set \mathcal{X} .

We achieve this data-dependence by making the parameters $\theta_{s;j}$ and $\theta_{st;jk}$ functions of x instead of constants, so

$$P_\theta(y|x) = P_{\theta(x)}(y) \propto \exp(-E_{\theta(x)}(y)) \quad (4)$$

with

$$E_{\theta(x)}(y) = \sum_{s \in V} \sum_{j \in \mathcal{L}} \theta_{s;j}(x) \mathbb{I}[y_s = j] + \sum_{(s,t) \in \mathcal{E}} \sum_{(j,k) \in \mathcal{L} \times \mathcal{L}} \theta_{st;jk}(x) \mathbb{I}[y_s = j \wedge y_t = k]. \quad (5)$$

Given a training set, $\{(x^1, y^1), (x^2, y^2), \dots, (x^T, y^T)\}$, of i.i.d. examples from an unknown distribution $d(x, y)$

this leads to the following learning problem: identify functions $\hat{\theta}_{s;j} : \mathcal{X} \rightarrow \mathbb{R}$ for $s \in V$ and $j \in \mathcal{L}$, as well as $\hat{\theta}_{st;jk} : \mathcal{X} \rightarrow \mathbb{R}$ for $(s, t) \in \mathcal{E}$ and $(j, k) \in \mathcal{L} \times \mathcal{L}$, such that for every x^i , $P_{\hat{\theta}(x^i)}(y)$ is an as good as possible estimate of the unknown conditional distribution $d(y|x^i)$.

Inspired by the data-independent case, we again first define the unary and pairwise marginals of the training examples. Since there is only one labeling of each training example, these are just indicators of the occurring labels,

$$\hat{\mu}_{s;j}(x^i) = \mathbb{I}[y_s^i = j], \quad \hat{\mu}_{st;jk}(x^i) = \mathbb{I}[y_s^i = j \wedge y_t^i = k] \quad (6)$$

for $i = 1, \dots, T$. We then learn (regression) functions

$$f_{s;j} : \mathcal{X} \rightarrow (0, 1) \subset \mathbb{R}, \quad f_{st;jk} : \mathcal{X} \rightarrow (0, 1) \subset \mathbb{R}, \quad (7)$$

that generalize these marginals, i.e. they approximately reproduce the values observed on the training data, but generalize to the complete domain \mathcal{X} . Subsequently, we continue analogously to the data-independent situation. We set

$$\hat{\theta}_{s;j}(x) = \log f_{s;j}(x), \quad \hat{\theta}_{st;jk}(x) = \log \frac{f_{st;jk}(x)}{f_{s;j}(x) f_{t;k}(x)}, \quad (8)$$

from which we obtain the conditional CRF distribution using Equation (4) and (5).

Overall, we have reduced the problem of maximum likelihood CRF learning to problem of learning a set of independent regression functions, one per edge (or edge type) and label combination (Algorithm 1). Note that for any node, s , that is part of at least one edge, (s, t) , we do not have to learn the unary functions, $f_{s;j}(x)$ by a regression steps. Instead, the marginalization condition, $\hat{\mu}_{s;j}(x^i) = \sum_{k \in \mathcal{L}} \hat{\mu}_{st;jk}(x^i)$, implies that we can obtain it from the pairwise functions as $f_{s;j}(x) = \sum_{k \in \mathcal{L}} f_{st;jk}(x)$. Thus, we train unary regression functions only for isolated nodes.

The main advantage of LS-CRF to classical maximum-likelihood or maximum-margin CRF training is its efficiency, since no probabilistic inference or energy minimization routines must be run over the training data in order to estimate parameters. Also, the regression problems rely only on observed data and have no dependencies between them, so they can be solved completely in parallel or even in a distributed way. The outcome, however, are not just independent per-variable score, but a proper energy function including pairwise terms that we can use for joint predictions by probabilistic inference or MAP prediction.

2.2. Parameterization

We illustrate two setup for learning the regression functions, (7), in practice: linear least-squares regression, and non-linear regression by gradient boosted decision trees.

To improve the readability, we drop the indices indicating the label pair, jk , from the notation, understanding that

Algorithm 1 LS-CRF – Training

input training data: $(x^i, y^i, G^i)_{i \in I}$ for $I = \{1, \dots, n\}$,
 x^i : images, y^i : ground truth, $G^i = (V^i, \mathcal{E}^i)$: graphs

- 1: set $\phi_{st}^i \in \mathbb{R}^D \leftarrow$ feature vector of edge $(s, t) \in \mathcal{E}^i$
- 2: **for** $j, k \in \mathcal{L} \times \mathcal{L}$ **do**
- 3: set $\mu_{st;jk}^i \leftarrow \mathbb{I}[y_s^i = j \wedge y_t^i = k] \quad \forall i \in I, (s, t) \in \mathcal{E}^i$
- 4: train f_{jk} from $\bigcup_{i \in I} \bigcup_{(s,t) \in \mathcal{E}^i} \{(\phi_{st}^i, \mu_{st;jk}^i)\}$
- 5: **end for**

output functions $f_{jk}: \mathbb{R}^D \rightarrow (0, 1)$ for $j, k \in \mathcal{L} \times \mathcal{L}$

all following steps have to be performed separately for each $(j, k) \in \mathcal{L} \times \mathcal{L}$. We furthermore assume, also for the sake of a more compact notation, that only one class of pairwise factors occur. This allows us to also drop the edge identifier, st , from the notation. As a consequence, multiple regions within each image will contribute to learning the same pairwise term. We write ϕ^1, \dots, ϕ^N for the feature representations of all such regions across all images, and μ^1, \dots, μ^N for their estimated marginals, (6), obtained from the ground truth annotation. Note that N is typically in the order of the total number of edges the graphs of all training images, so much larger than the number of training examples, n .

Linear Models. In the linear case, we parameterize² $f(x) = \langle w, \phi(x) \rangle$ and use least squares regression to obtain the weight vector, w ,

$$\min_w \quad \lambda \|w\|^2 + \sum_{i=1}^N \|f(x^i) - \hat{\mu}(x^i)\|^2, \quad (9)$$

where $\lambda \geq 0$ is an (optional) regularization constant. A major advantage of this formulation is that Equation (9) has a closed form solution for the optimal weight vector,

$$w = (\Phi\Phi^\top + \lambda I)^{-1} \Phi\mu \quad (10)$$

where Φ is the matrix with the features, ϕ^1, \dots, ϕ^N , as columns, and μ is the vector formed from the target outputs, μ^1, \dots, μ^N .

Computing Equation (10) for D -dimensional feature vectors requires solving a linear system of size $D \times D$. This is possible efficiently even for features that have several thousands dimensions, and using a precomputed LR-factorization of the matrix $(\Phi\Phi^\top + \lambda I)$, it is even possible to solve the regression problems for all label pair with minimal overhead compared to the time for a single pair. When the number of training examples is very large, however, it can happen that the computation of $\Phi\Phi^\top$ becomes the computational bottleneck. Minimizing (9) is still possible in this case using iterative least-squares solvers, such as the

²We suppress a possible bias term in the regression. It can be recovered by adding an additional, constant, entry to the feature representation.

Algorithm 2 LS-CRF – Prediction (Loopy case)

input image x , graph $G = (V, \mathcal{E})$

- 1: $\phi_{st} \in \mathbb{R}^D \leftarrow$ feature vector of edge $(s, t) \in \mathcal{E}$
- 2: **for** $j, k \in \mathcal{L} \times \mathcal{L}$ **do**
- 3: $\theta_{st;jk} = \log f_{jk}(\phi_{st}) \quad \forall (s, t) \in \mathcal{E}$
- 4: **end for**
- 5: $E(y) = \sum_{(s,t) \in \mathcal{E}, (j,k) \in \mathcal{L} \times \mathcal{L}} \theta_{st;jk} \mathbb{I}[y_s = j \wedge y_t = k]$

output energy function $E(y)$ for image x

method of conjugate gradients, or straight-forward stochastic gradient descent [4].

Non-Linear Models. From non-linearities in the regression functions we expect more flexible and better predictions. However, this applies only if enough data and computational resources are available to train them.

In this work, we use *gradient boosted regression trees* [10] as non-linear regressors. These have been shown to be strong predictors [6], while at the same time begin very fast to evaluate. The latter aspect is particularly relevant in the context of image segmentation, where many predictor evaluations are required for each image.

Tree and forest classifiers have a long tradition in image segmentation, but typically for different purposes. They are used either to construction per-(super)pixel feature representations or to predict per-(super)pixel unary classifier scores [11, 25, 24]. In both cases, however, it is still necessary to afterwards perform regular CRF learning.

LS-CRF, on the other hand, uses the trees to directly predict the coefficients of the energy function, replacing the need for additional CRF training that would require probabilistic inference. To our knowledge, the only exist method following a similar setup are *decision tree fields* (DTFs) [22]. These also learn the potentials of a random field using non-parametric decision trees. However, they differ significantly in their technical details: DTFs are trained using the pseudo-likelihood principle, and they parameterize the weights in a particular hierarchical way that is learned jointly with their values. LS-CRF, on the other hand, can be implemented using any out-of-the-box regression method, trees being just one particular choice.

2.3. Extension to loopy graphical models

It follows from the consistency of the maximum likelihood procedure that the construction we describe above leads to a consistent training algorithm for CRFs, as long as the underlying graphical model has tree topology. However, many interesting models in computer vision are loopy, in particular those used for image segmentation. In this section we describe how to construct an energy function in this situation that approximates the one one would obtain from

(intractable) maximum likelihood learning, and that works well in practice.

We first observe that the LS-CRF training procedure (Algorithm 1) can be performed regardless of whether the underlying model is loopy or not, since it does not require probabilistic inference at training time. At test time, however, we should not simply apply the rules (8), since the terms in the resulting energy would not balance in the correct way when the underlying graph has loops. Instead, we use the following construction (Algorithm 2): first, we decompose the (loopy) graph of a new image into a collection of subgraphs. This is inspired by tree-reweighted message passing [16], where each subgraph is a tree. In our case, we choose the simplest possible trees, i.e., single edges with their two adjacent vertices. For each such edge, (s, t) , we use the expressions for $\hat{\theta}_{s;j}(x)$ and $\hat{\theta}_{st;jk}(x)$ (Equation (3)) to obtain a partial energy function,

$$E_{\hat{\theta}(x)}^{(s,t)} = \sum_{(j,k) \in \mathcal{L} \times \mathcal{L}} E_{\hat{\theta}(x);jk}^{(s,t)} \mathbb{I}[y_s = j \wedge y_t = k], \quad (11)$$

with

$$\begin{aligned} E_{\hat{\theta}(x);jk}^{(s,t)} &= \hat{\theta}_{st;jk}(x) + \hat{\theta}_{s;j}(x) + \hat{\theta}_{t;k}(x) \\ &= \log \frac{f_{st;jk}(x)}{f_{s;j}(x)f_{t;k}(x)} + \log f_{s;j}(x) + \log f_{t;k}(x) \\ &= \log f_{st;jk}(x). \end{aligned} \quad (12)$$

Summing the expressions (11) over all edges we obtain a joint energy function, $E_{\hat{\theta}(x)}$ of all variables. It is determined completely by the pairwise regression functions that were learned from the training data.

Further extensions of LS-CRF are imaginable. For example, graphical models with higher order terms could be handled by the junction tree generalization of Equation (3) (see [30]). However, we leave the realization of such extensions to future work.

3. Related work

A large body of prior work exists on the topic of conditional random field training. In this section we discuss only some of the most related works. For a much broader discussion, see, e.g., the overview articles [8, 21, 30]. We also discuss our contribution only in context of probabilistic CRF training. Maximum-margin learning, in particular structured support vector machines (SSVMs) [28] rely on different assumptions and optimization techniques. For a comparison see, e.g., [20]. So far few decomposition-based techniques exist for SSVMs. An example is [17], which introduces a scheme of alternating between two steps: solving independent max-margin subproblems and updating Lagrange multipliers to enforce consistency between the solutions of the subproblems. However, as a first-order dual

decomposition technique, many iterations can be required until convergence, and experience in a large-scale setting is so far missing.

Traditional probabilistic CRF training aims at finding a weight vector, w , that maximizes the conditional likelihood of the training data under an assumed log-linear model for the conditional distribution, $P_w(y|x) \propto \exp(-E(x, y; w))$ with $E(x, y; w) = \langle w, \phi(x, y) \rangle$. Equivalently, one minimizes the negative conditional log-likelihood,

$$\ell(w) = \lambda \|w\|^2 - \sum_{i=1}^T E(x^i, y^i; w) + \log Z(x^i; w) \quad (13)$$

where $\lambda \geq 0$ is a regularization constant and $Z(x^i; w) = \sum_{y \in \mathcal{Y}} \exp(-E(x^i, y; w))$ is the *partition function*.

This optimization problem is smooth and convex so – in principle – gradient based optimization techniques, such as steepest descent, are applicable. In practice, however, the exponential size of \mathcal{Y} make this intractable, except in a few well-understood cases, such as model of very low tree width. Otherwise, computing Z or its derivatives is $\#P$ -hard [5], so already computing a single exact gradient of (13) is computationally intractable.

For tree-shaped models, the gradients can be computed in polynomial time. Nevertheless, minimizing (13) exactly is possible in practice only when the number of training images is small, since every gradient evaluation requires probabilistic inference across all training examples. *Stochastic gradient descent* (SGD) training has been proposed to overcome this [4, 29], but since it still requires probabilistic inference in the underlying graphical model its use is limited to small and tree-shaped models.

For larger or loopy models, approximate inference methods have been proposed that do not approximate the gradients of (13), but replace the whole objective by an easier one. Prominent examples are pseudolikelihood (PL) [3, 22] and piecewise (PW) training [25, 26]. These methods replace the log-likelihood, (13), by a surrogate that allows easier minimization, e.g., by bounding the intractable partition function, Z , by a factorized approximation.

LS-CRF shares several properties with PL and PW training methods, in particular the fact that can be phrased as solving a set of independent optimization problems and does not require probabilistic inferences at training time. However, it has some additional desirable properties shared by neither of the earlier techniques: in particular, only LS-CRF provides a closed form solution for the coefficient of the energy function. Also, both PL and PW are typically derived for log-linear conditional distributions, whereas LS-CRF make no assumption on the parametric form of the predictors, which makes it easy to train also non-linear models.

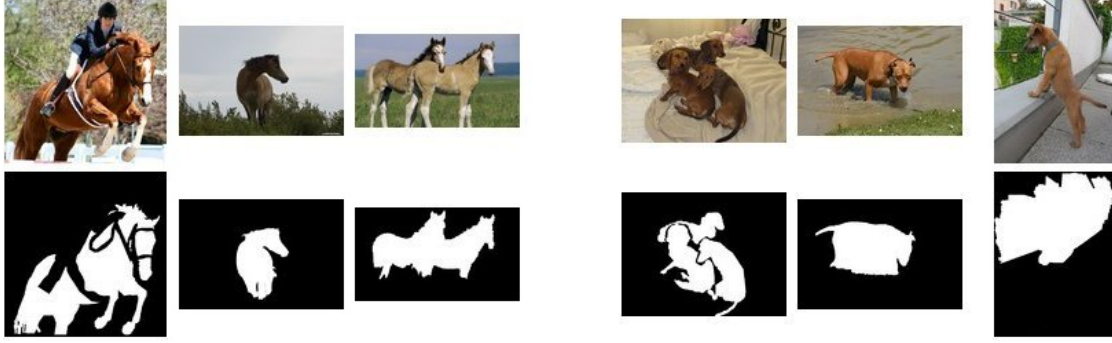


Figure 3: Example images and segmentation masks from *HorseSeg* (left) and *DogSeg* (right) datasets. For each dataset, we illustrate: manually created annotation (left), annotation from bound boxes (middle) and annotation from per-image labels (right). Annotation from bounding boxes is typically rather accurate, annotation from labels can contain significant errors.

4. Large-Scale Image Segmentation

Our main interest in the development of LS-CRF is the problem of large-scale image segmentation. The input data are RGB images of variable sizes and the goal is to predict a segmentation mask of the same size in which each pixels or superpixel is assigned one out of r values. In the case of *multi-class semantic segmentation*, these labels correspond to semantic classes, which can be ‘stuff’, such as *road*, or *sky*, or ‘things’, such as *cars* or *people*. The number of labels, r , is typically between 5 and 20 in this case. A second case of interest is *figure-ground segmentation*, where $r = 2$ and the labels indicate *foreground* and *background*.

4.1. Datasets

For our experiments on the multi-class semantic situation, we use the 8-label *Stanford background* dataset [12], which has been created by merging images of urban and rural scenes from four existing image datasets: LabelMe [23], MSRC [25], PASCAL VOC [9] and Geometric Context [13]. Despite this effort, the dataset contains only 715 images, so we do not consider it a large-scale dataset by current standards. In fact, none of the current datasets for natural image segmentation contains significantly more than a thousand images. The main reason is that providing pixel-wise ground truth annotation is time consuming and therefore costly, even when performed on a crowd sourcing platform like Amazon Mechanical Turk.

To overcome this limitation, we make a second contribution in this manuscript: two large-scale datasets for figure-ground image segmentation, *HorseSeg* with over 25,000 images and *DogSeg* with over 156,000 images. The images for both datasets were collected from the ImageNet dataset [7] and the *trainval* part of PASCAL VOC2012 [9]. As test set, we use 241 horse images and 306 dog images, which were annotated manually with figure-ground segmentations.

All images from the PASCAL dataset have manually created ground truth annotation. For the remaining images

from ImageNet, only manual annotation by bounding boxes or per-image labels is available. As such, the dataset provides a natural benchmark for large-scale, semi-supervised, image segmentation with different amount of ground truth.

To enable experiments on large-scale supervised training, we also provide semi-automatically created figure-ground annotation for all images of the two datasets, based on the following procedure. We first group the images by whether they have bounding box annotation for the foreground object available or not. For the annotated part (approximately 6,000 of the *horse* images and 42,000 of the *dog* images), we apply the *segmentation transfer* method of [18] to the bounding box region. A visual inspection of the output (see Figure 3) shows that the resulting segmentations are often of very high accuracy, so using them as a proxy for manual annotation seems promising. For the remaining images, only per-image annotation of the class label is known. To these images we apply the unconstrained segmentation transfer, which also yields figure-ground segmentation mask, but of mixed quality (see Figure 3). In Section 4 we report on experiments that use these different subsets for training CRF models. Note that even if the training data is generated by an algorithm, the evaluation is performed purely on manually created ground truth data, so the evaluation procedure is not biased towards the label transfer method.

4.2. Image Representations

We represent each image by a graph of SLIC superpixels [1] with an edge for any touching pair of superpixels. For each superpixel, s , we compute the following features:

- $\phi_s^{\text{color}} \in [0, 255]^3$: average RGB color in s ,
- $\phi_s^{\text{pos}} \in [0, 1]^2$: center of s in relative image coordinates,
- $\phi_s^{\text{sift}} \in [0, 16]^{128}$: rootSIFT descriptor [2] at center of s .
- $\phi_s^{\text{tree}} \in [0, 1]^8 / [0, 1]^{10}$: predicted label probabilities.

The ϕ_s^{tree} features are used for the linear regression and pseudolikelihood experiments. They consist of the outputs of per-label training boosted tree classifiers on a subset of the available training data. On the Stanford dataset this results in 8 additional feature dimensions. For DogSeg and HorseSeg, we use the Stanford features plus the output of per-superpixel classifiers of the foreground and background classes, resulting in overall 10 additional dimensions. For each edge (s, t) we define a feature representation by concatenating the features of the contributing superpixels, $\phi_{st} = [\phi_s, \phi_t]$.

4.3. Implementation

Training a CRF by LS-CRF requires only solving multiple regression problems, a task for which several efficient software packages are readily available. In our experiments with linear regression, we use the *Vowpal Wabbit* package³ (LBFGS optimization, learning rate 0.5, no explicit regularization), clamping the predictions to the interval $[10^{-9}, 1]$. *Vowpal Wabbit* is particularly suitable for large scale learning, since it supports a variety of infrastructures, from single CPUs to distributed compute clusters. As non-linear setup we train gradient boosted regression trees using the *MatrixNet* package [27] with default parameters (500 oblivious trees, depth 6, learning rate 0.1).

As baselines, we use CRFs with only unary terms, trained with the same regression methods as LS-CRF. We furthermore implemented a baseline that performs segmentation with only unary terms trained in the usual way with a logistic loss, and a CRF with pairwise terms using pseudolikelihood and piecewise training. Latter methods rely on the *grante* library⁴ and *Vowpal Wabbit* package. Other training techniques, in particular those requiring probabilistic inference, are not included, since they do not scale to the size of datasets we are interested in.

At test time, we use MAP prediction to create segmentations of the test images. For this, we need to minimize the energy function that the training methods have produced. In the unary-only case, this is possible on a pixel-by-pixel level. For energy function with pairwise terms (LS-CRF and pseudolikelihood), we use MQPBO [15] followed by TRWS [16] for the figure-ground segmentations. For both we use the implementations provided by the *OpenGM2* library⁵.

Alternatively, we could use solvers based on integer linear programming, which have recently been found effective for image segmentation tasks [14]. We plan to do so in future work.

Model	unary	pairwise
LS-CRF linear	70.3 (1.0)	73.4 (1.1)
LS-CRF non-linear	70.9 (1.1)	73.3 (1.1)
logistic regression	70.4 (0.8)	–
pseudolikelihood	–	71.0 (1.1)
piecewise	–	72.9 (0.9)

Table 1: Segmentation quality (average per pixel accuracy in %) on *Stanford background* dataset. Numbers in brackets are standard deviations over five cross-validation splits.

4.4. Results

We first report experiments on the multi-label Stanford Background dataset. While this is not a large-scale setup, the purpose is to show that LS-CRF achieves segmentation accuracy comparable to existing techniques for approximate CRF training, in particular as least as good as pseudolikelihood training, which is commonly used for this kind of problems. Afterwards, we report results in the large-scale regime, using the DogSeg and HorseSeg dataset with semi-automatic annotation.

Stanford Background Dataset. Table 1 summarizes the results on the Stanford Background dataset in numeric form. For example segmentations, please see the supplemental material. We compare seven setups: LS-CRF (unary-only or pairwise energies) with linear or non-linear parameterization, logistic regression (unary-only), pseudolikelihood and piecewise (pairwise only).

The results show that for learning a segmentation model with only unary terms, the squared loss objective of LS-CRF achieves comparable result to usual probabilistic loss (logistic regression). Including pairwise terms into the model improves the segmentation accuracy when we train the model with piecewise method or with LS-CRF. Pairwise terms trained by pseudolikelihood training have only a minor positive effect on the segmentation quality in our experiments. The non-linear and the linear versions of LS-CRF achieve comparable performance, likely because the amount of training data per label or label pair is small, so the classifier is not able to make use of the additional flexibility of a non-linear decision function. In fact, approximately 20 of the 64 possible label combination occur too rarely in the training images to train predictors for them. Instead, we assigned them constant probabilities of 10^{-3} .

Note that the numbers for pairwise models are also roughly comparable to the 74.3% average per pixel accuracy reported in the original work [12] for a CRF with pairwise term. We attribute the existing difference in values to our use of simpler (and faster computable) image features.

DogSeg/HorseSeg dataset. In the large scale regime, we are interested in three questions. Can we make truly large

³<http://hunch.net/~vw/>

⁴<http://www.nowozin.net/sebastian/grante/>

⁵<http://hci.iwr.uni-heidelberg.de/opengm2/>

(a) images with manual annotation

	HorseSeg [147]		DogSeg [249]	
Model	unary	pairwise	unary	pairwise
linear	81.4 ($<1m$)	82.2 ($<1m$)	77.8 (2.5m)	79.1 (2.5m)
non-linear	81.6 ($<1m$)	83.8 (1.5m)	78.5 ($<1m$)	80.8 (2m)

(b) images with manual or object bounding box annotation

	HorseSeg [6044]		DogSeg [42763]	
Model	unary	pairwise	unary	pairwise
linear	82.0 ($<1m$)	83.6 ($<1m$)	78.5 (7m)	81.2 (14m)
non-linear	83.6 (9m)	86.4 (32m)	80.9 (110m)	83.8 (348m)

(c) all images

	HorseSeg [24837]		DogSeg [156062]	
Model	unary	pairwise	unary	pairwise
linear	81.4 (1m)	83.3 (2m)	78.1 (20m)	80.2 (46m)
non-linear	82.5 (88m)	84.5 (354m)	80.0 (519m)	82.2 (668m)

Table 2: Results of LS-CRF for figure-ground segmentation on HorseSeg/DogSeg (average per class accuracy in %) for different training subsets. The numbers in brackets indicate training time and numbers in rectangular brackets indicate the number of training images.

scale CRF training feasible? How do CRFs benefit from the availability of large amounts of training data? How useful is annotation that was created semi-automatically?

We study these question in three sets of experiments, using different subsets of the training data: (a) only images with manually created annotation, (b) images with annotation created manually or by segmentation transfer using bounding box information, (c) all training images.

We train CRFs with pairwise terms using the linear and non-linear variant of LS-CRF, and compare their segmentation performance to models with only unary terms. In situation (a), we use the feature vectors of all available superpixels to train the unary-only models, and all neighboring superpixel pairs to train LS-CRF with pairwise terms. In the larger setup, (b) and (c), we reduce the redundancy in the data by using only 25% of all superpixels for the unary-only models, sampled in a class-balanced way. For pairwise models, we record the ratio of pairs with *same label* versus with *different label*. Preserving this ratio, we sample 10% of all superpixels pairs, in a way that combinations with both *foreground* and both *background* are equally likely, and also *foreground/background* and *background/foreground* transitions are equally likely. The per-

centages are chosen such that the training problems in both situations are of comparable size. On the *DogSeg* dataset, they consists of approximately 90K data points for situation (a), 3.5M data points for (b), and 13M data points for (c), except the pairwise/nonlinear case, where we use only 6.5M data points for memory reasons. On the *HorseSeg*, the number are roughly half as big for (a), and one sixth for (b) and (c).

A first observation is that using LS-CRF training is computationally feasible even in the largest setup: for example, training with all training images in the *DogSeg* dataset with linear objective with pairwise terms required 45 minutes on a 24 core workstation, compared to 20 minutes, if only unary terms are learned. The non-linear setup took approximately 9 hours to train an energy function with only unary terms, and 11 hours when pairwise terms were used.

Table 2 shows numeric results for the segmentation accuracy and training time of the different setups. Example segmentation are provided in the supplemental material. The results allow us to make several observations that we believe will generalize beyond the specific setup of our work.

First, it has been observed previously that pairwise terms often have only a minor positive effect on the segmentation quality (e.g. [11]). Our experiments show a similar trend when a linear representation was used and the number of training examples was small. However, when a large training set was used, the difference between unary-only and pairwise models increased. Second, the use of non-linear predictors consistently improved the segmentation quality. This could be a useful insight also for other CRF training methods, which rely predominantly on linearly parameterized energy functions. Third, the segmentation quality improved significantly when increasing the number of training images, even though the additional images had only annotation created automatically using information from bounding boxes or per-image labels. This indicates that segmentation transfer followed by large-scale CRF learning could be a promising way for leveraging the large amounts of unlabeled image data, e.g. in the Internet.

One can also see in Table 2 that the segmentation accuracy was highest when training on the manually annotated image set together with images that had bounding boxes annotation. Including also the remaining images led to a reduction in the quality. This raises the question whether the annotation created only from per-image labels contain useful information at all. We performed additional experiments for this, measuring the segmentation quality of training linear LS-CRF on the HorseSeg dataset, but training exclusively on images with annotation from bounding boxes, or on images with annotation from per-image labels. This resulting per-class accuracies are 82.0 (unary) and 83.9 (pairwise) for the bounding box case, and 81.1 (unary) and 82.8 (pairwise) for the per-image case. Comparing this to the

values 81.4 (unary) and 82.2 (pairwise) from Table 2, we see that the automatically generated segmentations do indeed contain useful information. Training only on these achieves results comparable to training on the (admittedly much fewer) manually annotated images.

5. Summary

In this work we make two main contributions:

- a new technique for inference-free CRF training that scales to very large training sets,
- two new benchmark datasets of over 180,000 images and segmentation masks.

We used these to perform the first truly large-scale experiments in the area of (semantic) image segmentation, which provided us with several noteworthy observations: 1) the positive effect of pairwise terms increased with the number of training examples, 2) training CRFs with non-linear energies is feasible and results in better segmentation models, 3) semi-supervised learning is practical and useful also for image segmentation, by (semi-)automatically generation annotation for otherwise unlabeled images.

Furthermore, we are convinced that both contributions will be useful also outside of the area of image segmentation. The large-scale CRF training method is applicable regardless of the application area, and the dataset can serve also as a generic testbed for large scale training of looped CRF models.

References

- [1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *PAMI*, 34(11), 2012. 6
- [2] R. Arandjelović and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, 2012. 6
- [3] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 1975. 5
- [4] L. Bottou. Large-scale machine learning with stochastic gradient descent. In *International Conference on Computational Statistics (COMPSTAT)*, 2010. 4, 5
- [5] A. Bulatov and M. Grohe. The complexity of partition functions. *Theoretical Computer Science*, 348(2), 2005. 5
- [6] R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *ICML*, 2006. 4
- [7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 6
- [8] J. Domke. Learning graphical model parameters with approximate marginals inference. *PAMI*, 2013. 5
- [9] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The Pascal visual object classes (VOC) challenge. *IJCV*, 88(2), 2010. 6
- [10] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 2001. 4
- [11] B. Fulkerson, A. Vedaldi, and S. Soatto. Class segmentation and object localization with superpixel neighborhoods. In *ICCV*, 2009. 4, 8
- [12] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, 2009. 6, 7
- [13] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 75(1), 2007. 6
- [14] J. H. Kappes, B. Andres, F. A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B. X. Kausler, J. Lellmann, N. Komodakis, and C. Rother. A comparative study of modern inference techniques for discrete energy minimization problem. In *CVPR*, 2013. 7
- [15] P. Kohli, A. Shekhovtsov, C. Rother, V. Kolmogorov, and P. Torr. On partial optimality in multi-label mrfs. In *ICML*, 2008. 7
- [16] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28(10), 2006. 5, 7
- [17] N. Komodakis. Efficient training for pairwise or higher order CRFs via dual decomposition. In *CVPR*, pages 1841–1848, 2011. 5
- [18] D. Küttel, M. Guillaumin, and V. Ferrari. Segmentation propagation in ImageNet. In *ECCV*, 2012. 6
- [19] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001. 1
- [20] S. Nowozin, P. V. Gehler, and C. H. Lampert. On parameter learning in CRF-based approaches to object class image segmentation. In *ECCV*, 2010. 5
- [21] S. Nowozin and C. H. Lampert. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6, 2011. 1, 5
- [22] S. Nowozin, C. Rother, S. Bagon, T. Sharp, B. Yao, and P. Kohli. Decision tree fields. In *ICCV*, 2011. 4, 5
- [23] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *IJCV*, 77(1-3), 2008. 6
- [24] F. Schroff, A. Criminisi, and A. Zisserman. Object class segmentation using random forests. In *BMVC*, 2008. 4
- [25] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *IJCV*, 81(1), 2009. 4, 5, 6
- [26] C. Sutton and A. McCallum. Piecewise training of undirected models. In *UAI*, 2005. 5
- [27] I. Trofimov, A. Kornetova, and V. Topinskiy. Using boosted trees for click-through rate prediction for sponsored search. In *International Workshop on Data Mining for Online Advertising and Internet Economy*, 2012. 7
- [28] I. Tsochantaris, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6, 2005. 5
- [29] S. Vishwanathan, N. N. Schraudolph, M. W. Schmidt, and K. P. Murphy. Accelerated training of conditional random fields with stochastic gradient methods. In *ICML*, 2006. 5
- [30] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 2008. 2, 3, 5

6. Supplementary material



Figure 4: Image segmentation examples from Stanford background dataset for pairwise non-linear model.



Figure 5: Test image segmentation examples from HorseSeg dataset for pairwise non-linear model trained on images with manual annotation or object bounding box (green color means foreground, red — background).

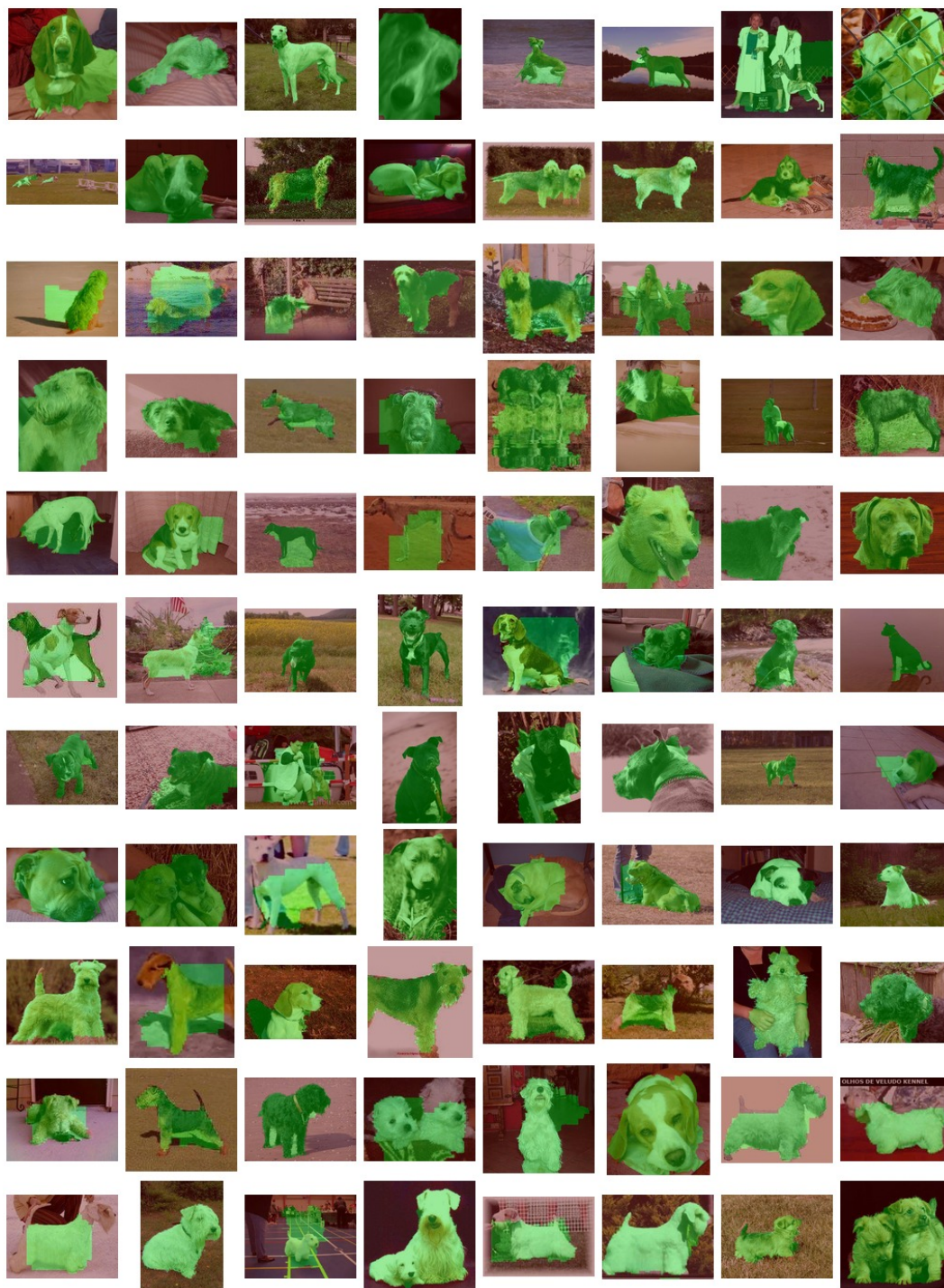


Figure 6: Test image segmentation examples from DogSeg dataset for pairwise non-linear model trained on images with manual annotation or object bounding box (green color means foreground, red — background).



Figure 7: HorseSeg training image examples.



Figure 8: DogSeg training image examples.